Deployment Strategies for Differentiated Detection in Wireless Sensor Networks

Jingbin Zhang, Ting Yan, and Sang H. Son
Department of Computer Science
University of Virginia, Charlottesville 22903
{jz7q, ty4k, son}@cs.virginia.edu

Abstract— In this paper, we address the deployment problem for differentiated detection requirements, in which the required detection probability thresholds at different locations are different. We focus on differentiated deployment algorithms that are applied to the probabilistic detection model, since it is more realistic than the binary detection model. We show that the relationship between the node deployment strategy and the logarithmic collective miss probability distribution is Linear Shift Invariant (LSI). Using this property, we formulate the differentiated deployment problem as an integer linear programming problem, which is a well known NP-hard problem. We propose a differentiated node deployment algorithm called DIFF_DEPLOY, which achieves much better performance than the state-of-the-art node deployment algorithm for both uniform and differentiated detection requirements.

I. INTRODUCTION

The emergence of wireless sensor networks gives rise to many applications, such as military surveillance [1] and habitat monitoring [2]. In these applications, it is an important requirement to provide adequate sensing coverage to achieve acceptable quality of service.

In previous papers on sensing coverage [3][4][5][6], a binary detection model is assumed. In a binary detection model, sensor node can detect a target with a 100% probability provided that the target is within its sensing range, and cannot detect a target beyond the range. In this paper, we use a probabilistic detection model, because in reality the detection of a target is not deterministic due to the uncertainty associated with sensor detections. When a target is within a sensor's sensing range, whether the target is detected by the sensor is probabilistic.

With a probabilistic detection model, we can hardly have a 100% detection probability for all the geographic points in the target area. In many surveillance applications, the system requires different degrees of surveillance at different locations. The system may require extremely high detection probabilities at certain sensitive areas. However, for some not so sensitive areas, relatively low detection probabilities are required to reduce the number of sensors deployed so as to decrease the cost. In this case, different areas require different densities of deployed nodes. When the detection probability thresholds at different subareas are specified, the minimum number and the deployment locations of the sensors need to be decided based on the specification and the probabilistic detection model. We name it as the "Differentiated Deployment" problem. The goal of the paper is to develop a deployment strategy to satisfy the different detection probability thresholds at different locations using minimum number of nodes.

Minimizing the number of nodes deployed may not be critical when the cost of node is negligible and we have excessive number of nodes. However, as long as the prices of sensor nodes are not negligible, to reduce the number of nodes deployed is always necessary. For example, while MicaZ motes, which are commonly used in terrestrial environments, cost over 120 dollars each [7], some sensors used in undersea surveillance applications are tens of or even hundreds of times more expensive. While it is expected that the cost of sensors will decrease as the technologies advance, it may not happen quickly, since the advances of the technologies also result in more powerful features being incorporated into a single node. This is especially true for undersea sensor nodes. Another issue is stealthiness. By minimizing the number of sensors deployed, the risk of the system being detected by adversaries is also reduced. Even if the cost and the stealthiness are not an issue, knowing the minimal number and the deployment locations of the nodes provides us the guidance on how to deploy the redundant sensors to improve reliability. For example, the redundant sensors can be deployed proportionally, based on the density of the sensors after a minimum number of sensors are deployed.

We assume that we have good control of the node deployment, i.e., we can place the sensors into the exact targeted locations, either manually [8] or air-dropped [9]. However, when spatial error is inevitable in the node deployment, we can model the uncertainty of nodes' actual locations using Gaussian distribution [10]. We show that to consider uncertainty in the node deployment, we only need to modify the node detection model. Therefore, our algorithm is applicable to both precise node deployment and node deployment with uncertainty.

In this paper, we show that the relationship between the node deployment strategy and the logarithmic collective miss probability distribution of the sensor field is Linear Shift Invariant (LSI). By taking advantage of this property, we formulate the node deployment problem as an integer linear programming problem, which is a well known NP-hard problem. Further, based on the linear relationship, we devise a differentiated deployment algorithm called DIFF_DEPLOY, which outperforms the state-of-the-art probability based node

Report Documentation Page	Form Approved OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE **Deployment Strategies for Differentiated Detection in Wirelesss Sensor Networks**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Virginia,Department of Computer Science,151 Engineer's Way,Charlottesville,VA,22904-4740**	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **10**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

deployment algorithm MIN_MISS for both uniform and differentiated detection requirements.

The remainder of the paper is organized as follows. We review related work in Section II. We describe the sensor and terrain models in Section III. Section IV presents how to relate the detection probability to system performance. We formulate the node deployment problem in Section V. In Section VI, we present our differentiated deployment algorithm DIFF_DEPLOY. Then we briefly describe MIN_MISS and the way uncertainty is handled in Section VII. Section VIII contains a complete evaluation of different deployment algorithms. We present the conclusions and future work in Section IX.

II. RELATED WORK

In sensor network based surveillance applications, it is critical to determine where to deploy the sensors to satisfy the sensing coverage requirement. A large number of publications exist in the literature [10][3][11][4][5][12][6][13]. However, most of them are based on binary detection models [3][4][5][6].

When the binary detection model is used, if we only consider sensing coverage, to determine the minimum number of nodes to cover the whole area is actually a circle covering problem [5][14]. A lot of work also considers communication connectivity requirements while minimizing the number of nodes deployed [5][6].

Although the binary detection model simplifies the analysis, it is not realistic in many cases. In reality, a sensor detection is a probabilistic event, in which a target is not always detected by a sensor due to its limited sensing capability and processing power. In [12], Meguerdichian et. al. assume that the sensor coverage decreases as the distance from the sensor increases and propose an algorithm based on Voronoi diagram and Delaunay triangulation to find the maximal breach path and maximal support path. Although that algorithm can be used as a guide to deploy extra sensors to improve breach coverage, it can not be used for differentiated deployment problem unless both different node detection models and differentiated detection requirements are considered. In [11], Dhillon et. al. propose to use probabilistic detection models to better address this problem. In these models, the probability of target detection by a sensor decreases exponentially when the distance between them increases.

Several deployment algorithms [10][11][13] are proposed when the probabilistic detection models are used. The algorithm described in [13] assumes the mobility of the sensors, while we consider static sensors that do not move once they are deployed. The deployment algorithms described in [10][11] are similar, between which MIN_MISS proposed in [10] is more sophisticated. MIN_MISS is mainly used for uniform detection requirements, in which the specified detection probability threshold for the entire surveillance area is the same. Different from MIN_MISS, DIFF_DEPLOY is designed for both uniform and differentiated detection requirements. Furthermore, DIFF_DEPLOY achieves much better performance in both cases.

III. SENSOR AND TERRAIN MODELS

In this paper, we use a two-dimensional grid to cover the sensor field. We only consider the detection probability at the grid points. The granularity of the grid is adjusted based on the precision we want and the time and space we can afford for computation. Suppose the dimension of the grid is $U \times V$. The deployment strategy of the sensor field can be represented by a $U \times V$ matrix D, in which $D(x,y)$ denotes the number of nodes deployed at grid point (x,y). If $D(x,y) = 1$, it means that one sensor is deployed at grid point (x,y). If $D(x,y) = 0$, it means no sensor is deployed at that location. Typically, $D(x,y)$ is either 0 or 1. However, in very rare cases when the detection requirement is so high that more than one node need to be deployed in a small neighborhood, it is possible that $D(x,y)$ takes a value greater than 1. We use $d((m,n),(i,j))$ to denote the Euclidean distance between locations (m,n) and (i,j).

In reality, it is challenging to obtain an exact and precise node detection model, because the detection of a target depends on many factors. We classify the factors into three categories. Factors in the first category are those related to the sensors, such as the type and the quality of the sensor, and the sensing algorithm used to detect the target. Factors in the second category are those related to the target. They include: the type, size and shape of the target; the duration of a target staying at a certain location, which is related to the speed of the target; the distance between the target and the sensor. Factors in the third category are those related to the environment where sensors are deployed. This category includes the following factors: the surrounding obstacles; the weather, such as the temperature, humidity and whether it is rainy or windy; background noise, such as the magnetic field of the earth. Almost all these factors are experienced in real field experiments [15]. Many of these factors are hard to model. Therefore, it is very difficult to get a totally accurate detection model.

However, it is possible to obtain a conservative detection model, as what has been done to obtain a conservative sensing range [16]. This kind of model can be obtained by using conservative values in theoretical analysis or through extensive experiments. For example, we can run the experiments many times under different weather conditions, and use the lowest detection probability if the results under different weather conditions are different. In this way, we rule out several hard to model factors and the obtained model provides the worst case detection probability. Although it may be time consuming, it is possible to get a conservative detection model with current technologies. Obtaining the detection model is not only useful for the node deployment problem, but helps us better understand the performance of the system. Providing a totally accurate detection model is out of the scope of this paper.

In [10][13], a probabilistic node detection model considering the factor of the distance between the target and the sensor

is used, which is shown in Equation (1):

$$p((m,n),(i,j)) =
\begin{cases}
e^{-ad((m,n),(i,j))} & \text{if } d((m,n),(i,j)) <= r_s \\
0 & \text{if } d((m,n),(i,j)) > r_s
\end{cases} \quad (1)$$

In the above equation, $p((m,n),(i,j))$ is the probability of a target at grid point (m,n) being detected by the node at grid point (i,j), r_s is the sensing range, and a is a parameter related to the physical characteristics of the sensing device, which can be obtained from field experiments. This model assumes that the detection probability varies exponentially with the distance between the target and the sensor. Although this model is not accurate, it reasonably characterizes the behavior of range sensing devices, such as infrared and ultrasound sensors [17]. This equation does not take into consideration of the time duration a target stays at a certain location. In reality, if a target stays at a certain location for a longer time, it results in a higher probability of being detected. Therefore, we incorporate the factor of the duration into the node detection model:

$$p((m,n),(i,j),t) =
\begin{cases}
1 - (1 - p((m,n),(i,j),1))^{\lceil t/T \rceil} & \text{if } t > T \\
p((m,n),(i,j)) & \text{if } 0 < t \leqslant T
\end{cases} \quad (2)$$

In this model, we use discrete time. The time unit T is the time period during which the sensing algorithm is executed and a decision is made on whether a target is detected. t denotes the duration the target stays at location (i,j). This model assumes that the target detection in different time units is independent. We also assume that $p((m,n),(i,j),1) = p((m,n),(i,j))$.

Note that the detetion probability at a certain location for one time unit can be easily converted to the detection probability for multiple time units. If the detection probability for one time unit is w, the detection probability for l time units is $1 - (1 - w)^l$. For simplicity, we assume that the detection probability thresholds specified by the users are the probabilities for one time unit. To be consistent with the detection probability thresholds, all the probabilities mentioned in the following sections are the probabilities for one time unit. Since $p((m,n),(i,j)) = p((m,n),(i,j),1)$, we directly use $p((m,n),(i,j))$ instead of $p((m,n),(i,j),1)$ for brevity.

Similar to $p((m,n),(i,j))$, $p_{miss}((m,n),(i,j))$ is used to denote the probability of a target at grid point (m,n) being missed by the node at grid point (i,j), which is shown in Equation (3):

$$p_{miss}((m,n),(i,j)) = 1 - p((m,n),(i,j)) \quad (3)$$

We use a $U \times V$ matrix M to denote the collective miss probability distribution of the whole field. The collective miss probability $M(x,y)$ means the probability of a target at grid point (x,y) being missed by all the sensors deployed in the field, which is given by:

$$\begin{aligned}
M(x,y) &= \prod_{(i,j) \in Grid} p_{miss}((x,y),(i,j))^{D(i,j)} \\
&= \prod_{(i,j) \in Grid} (1 - p((x,y),(i,j)))^{D(i,j)}
\end{aligned} \quad (4)$$

$Grid$ in Equation (4) denotes the whole sensor field where we want to deploy the sensors.

IV. DETECTION PROBABILITY vs SYSTEM PERFORMANCE

In the differentiated deployment problem, realizing the relationship between the detection probability and the system performance helps the designer to specify the detection probability thresholds based on the different performance requirements at different locations. In this section, we briefly discuss how to infer the system performance from the detection probability and vice versa.

In surveillance applications, primary performance requirements include detection performance and tracking performance. The detection probability is closely related to both of them. The detection performance is about whether a target can be detected when it appears in the network, and the detection delay if the target is detected. The tracking performance represents how well the target's trajectory is obtained, given that the target is detected.

For different target types, we use different requirements. We divide the targets into two categories: static targets and moving targets. For the first category, we are mainly interested in the detection performance. The detection probability at a certain location indicates whether a target can be detected if the target stays there for a certain period of time. We use $p_{detect}(x,y)$ to denote the detection probability at grid point (x,y) when the target stays at (x,y) for one time unit. If $p_{detect}(x,y) = w$, we know that if a target stays at (x,y) for l time units, the probability of being detected is $1 - (1 - w)^l$. The detection probability also indicates the detection delay. If $p_{detect}(x,y) = w$, with $c\%$ confidence level, the detection delay is within $\ln(1 - c\%)/\ln(1 - w)$ time units. For example, if $p_{detect}(x,y) = 0.5$, the detection delay is within 5 time units with 95% confidence level. If the system requirement is specified by the desired detection performance, the designer can compute the detection probability thresholds at different locations based on the relationship between the detection probability and the detection performance.

For moving targets, we are interested in both the detection performance and the tracking performance. Given the node detection model for moving targets, the relationship between the detection performance and the detection probability for moving targets is the same as that for static targets. The tracking performance mainly depends on the frequency of the detection reports from the network. A finer grained tracking is achieved if the time interval between the consecutive detection reports is smaller. The frequency of the detection report is indicated by the detection probability. If $p_{detect}(x,y) = w$, the average time interval between the consecutive detection reports at (x,y) is $1/w$. For example, if the detection probability in an

area is 0.5, the network on average generates detection reports every two time units when a target appears in that area. In this case, we are informed the locations of the target for half of the time, but we do not exactly know where the target is for the other half of the time. However, if the detection probability in the area is 0.99, we get the detection reports almost every time unit. In this case, we obtain a finer grained trajectory of the target, together with the duration of the target associated with each location of the trajectory. If the desired tracking performance at different locations is provided, the designer can obtain the detection probability threshold at each location based on the relationship between the detection probability and the tracking performance.

V. Problem Formulation

In this section, we formulate the node deployment problem as an integer linear programming problem, which has been proven to be NP-hard. Note that both the problem formulation and our differentiated deployment algorithm are not restricted to the detection model we use. While we use the detection model described by Equation (1), any detection model can be applied. We first assume that the node detection model remains the same no matter where the node is deployed. A solution is provided when different locations result in different node detection models, which is mainly caused by terrain complexities, such as obstacles in the environment.

In this paper, we study the system that transforms the node deployment strategy D (the input of the system) to the logarithmic collective miss probability distribution $\ln M$ (the output of the system). The input-output relationship of the system can be characterized by the node detection model $p((m,n),(i,j))$. We will prove later in this section that this system is a Linear Shift Invariant (LSI) system. From the property of the LSI system, the logarithmic collective miss probability distribution is the two-dimensional convolution of the node deployment strategy and the impulse response of this system, which is characterized by the node detection model.

In order to simplify the relationship among the node deployment strategy D, the node detection model $p((m,n),(i,j))$ and the collective miss probability distribution M, we first replace $p((m,n),(i,j))$ with $p_{miss}((m,n),(i,j))$. As shown in Equation (4), the collective miss probability $M(x,y)$ is the product of all the sensors' probabilities to miss the target at location (x,y). We use logarithm to convert the multiplicative relationship shown in Equation (4) to an additive relationship shown in Equation (5).

$$\ln M(x,y) = \sum_{(i,j)\in Grid} [D(i,j) \times \ln p_{miss}((x,y),(i,j))] \quad (5)$$

For simplicity, we use I to denote the logarithmic collective miss probability distribution $\ln M$. It is easy to see that $M(x,y) = e^{I(x,y)}$ for any $(x,y) \in Grid$. Through the conversions, the system described in Equation (5) is indeed a Linear Shift Invariant System (LSI), which takes D as the input and I as the output. This is shown by the following theorem.

Theorem 1: The system described in Equation (5) that transforms the node deployment strategy D to I is a Linear Shift Invariant (LSI) System.

Proof: We prove that the system described in Equation (5) has the three properties of an LSI system as follows:

1. Scaling. Considering two inputs D_1 and D_2, and $D_2 = \lambda D_1$, in which λ is an arbitrary nonnegative integer constant. Suppose I_1 is the output of D_1 and I_2 is the output of D_2. Then for any $(x,y) \in Grid$, we have:

$$
\begin{aligned}
I_2(x,y) &= \sum_{(i,j)\in Grid} [D_2(i,j) \times \ln p_{miss}((x,y),(i,j))] \\
&= \lambda \sum_{(i,j)\in Grid} [D_1(i,j) \times \ln p_{miss}((x,y),(i,j))] \\
&= \lambda I_1(x,y)
\end{aligned}
$$

Therefore, $I_2 = \lambda I_1$ and the system conforms to the scaling property.

2. Superposition. Assume that the outputs of two arbitrary inputs D_1 and D_2 are I_1 and I_2, correspondingly. Suppose I' is the output $D_1 + D_2$. Then for any $(x,y) \in Grid$, we have:

$$
\begin{aligned}
I'(x,y) &= \sum_{(i,j)\in Grid} [(D_1(i,j) + D_2(i,j)) \times \ln p_{miss}((x,y),(i,j))] \\
&= \sum_{(i,j)\in Grid} [D_1(i,j) \times \ln p_{miss}((x,y),(i,j))] \\
&+ \sum_{(i,j)\in Grid} [D_2(i,j) \times \ln p_{miss}((x,y),(i,j))] \\
&= I_1(x,y) + I_2(x,y)
\end{aligned}
$$

Therefore, $I' = I_1 + I_2$ and the system conforms to the superposition property.

3. Shift Invariant. Assume the output of an arbitrary input D is I. Shift the input D by m in the x-axis and n in the y-axis to form a new input D', in which $D'(x,y) = D(x-m,y-n)$ and m and n are arbitrary integers. Denote the corresponding output of D' by I'. Then for any $(x,y) \in Grid'$, in which $Grid'$ is the sensor field after we shift the original sensor field $Grid$ by m in the x-axis and n in the y-axis, we have:

$$
\begin{aligned}
I'(x,y) &= \sum_{(i',j')\in Grid'} [D'(i',j') \times \ln p_{miss}((x,y),(i',j'))] \\
&= \sum_{(i',j')\in Grid'} [D(i'-m,j'-n) \times \ln p_{miss}((x,y),(i',j'))] \\
&= \sum_{(i',j')\in Grid'} [D(i'-m,j'-n) \\
&\quad \times \ln p_{miss}((x-m,y-n),(i'-m,j'-n))] \\
&= \sum_{(i,j)\in Grid} [D(i,j) \times \ln p_{miss}((x-m,y-n),(i,j))] \\
&= I(x-m,y-n)
\end{aligned}
$$

Therefore, the system conforms to the shift invariant property.

From the above three properties, we have shown that the system is an LSI system. ∎

Since it is an LSI system, we use the impulse response to characterize how an input D is transformed into the output I.

We use g to denote the impulse response in this LSI sytem. When we use the node detection model shown in Equation (1), $g(x, y)$ is expressed in the following equation:

$$
\begin{aligned}
g(x, y) &= \ln p_{miss}((x, y), (0, 0)) \\
&= \begin{cases} \ln(1 - e^{-a\sqrt{x^2+y^2}}) & \text{if } \sqrt{x^2 + y^2} <= r_s \\ 0 & \text{if } \sqrt{x^2 + y^2} > r_s \end{cases}
\end{aligned}
\quad (6)
$$

Further, as an LSI system, the output I can be represented by the convolution result of the input D and the impulse response g, which is shown in Equation (7).

$$
\begin{aligned}
I(x, y) &= D(x, y) * g(x, y) \\
&= \sum_i \sum_j D(i, j) \times g(x - i, y - j)
\end{aligned}
\quad (7)
$$

Two dimensional convolution is a well studied area in image processing. We apply the same techniques used in image processing to transform the convolution to the matrix multiplication by constructing the corresponding matrices [18][19]. We do not discuss the details of how to construct the corresponding matrices in this paper due to the page limit. Readers are referred to [18][19] for details. By using the corresponding matrices, Equation (7) is represented by the following matrix Equation:

$$ I_p = G_p \cdot D_p \quad (8) $$

In Equation (8), matrix G_p is constructed based on g, matrix D_p is constructed based on D and matrix I_p is constructed based on I. By using an LSI system to represent the relationship among the node deployment strategy D, the node detection model $p((m, n), (i, j))$ and the collective miss probability distribution M, we finally reach a simple equation which characterizes the relationship among all three elements. Note that if $N = max(U, V)$, both the dimensions of I_p and D_p are $N^2 \times 1$, and the dimension of G_p is $N^2 \times N^2$. Also note that if the indices in G_p start from zero, the elements in G_p can be interpreted in the following way: the value of $G_p(x, y)$ equals to the value of $\ln p_{miss}((\lfloor x/N \rfloor, x\%N), (\lfloor y/N \rfloor, y\%N))$.

By now, we assume that all the nodes share the same detection model. However, this assumption does not hold in certain situations. For example, the sensing capability of a node may be undermined by the obstacles in the environment. This problem is solved by modifying the matrix G_p. For example, if a node at grid point (m, n) has a detection model different from others, we change the values of the elements in the $(m \times N + n)$th column of G_p. If the value of $p_{miss}((x, y), (m, n))$ is changed to be v_{new}, $G_p((x \times N + y), (m \times N + n))$ is set to $\ln v_{new}$. In this way, Equation (8) allows different detection models at different locations and the impact of obstacles is incorporated. By allowing different detection models at different locations, the system is no longer an LSI system. However, it is still a linear system which can be proven in the same way as the proof of the scaling and superposition properties of Theorem 1. So Equation (8)

still holds and both our problem formulation and differentiated node deployment algorithm are not affected.

Based on Equation (8), we are able to formulate the node deployment problem as an integer linear programming problem. We use m_{th} to denote the miss probability threshold distribution of the whole field, in which $m_{th}(x, y)$ is the miss probability threshold at location (x, y). Our objective is to find the minimal number of nodes to satisfy that, after these nodes are deployed, for any $(x, y) \in Grid$, the collective miss probability $M(x, y)$ is smaller than or equal to $m_{th}(x, y)$. We set I to $\ln m_{th}$. Then we construct the corresponding matrices for D, g and I. The problem can be formulated as follows:

Minimize: $sum(D_p)$
Subject to: $G_p \cdot D_p <= I_p$
The elements in D_p are nonnegative integers

Fig. 1.   Integer linear programming model.

This problem is a typical integer linear programming problem, which has been proven to be NP-hard. An NP-hard problem indicates that it is almost impossible to find the optimal solution when the input size is large, since we can not afford the computation time. Therefore, we devise an optimization algorithm called DIFF_DEPLOY to obtain the suboptimal result.

VI. DIFFERENTIATED DEPLOYMENT ALGORITHM

In this section, we present the differentiated deployment algorithm DIFF_DEPLOY. DIFF_DEPLOY makes use of the Linear Shift Invariant (LSI) property, which justifies the representation of the relationship among the node deployment strategy, the node detection model and the collective miss probability distribution by matrix multiplication as shown in Equation (8). Based on matrix algebra, it is clear that if we know the miss probability threshold distribution and the node detection model, we can directly compute the deployment strategy D_p based on Equation (8) as follows:

$$ D_p = G_p^{-1} \cdot I_p \quad (9) $$

The result D_p computed from Equation (9) indicates the number of sensors we should place at each location to satisfy that $M = m_{th}$, based on the node detection model. If we could place the sensors according to the computed result D_p, the total number of nodes deployed would be minimal, because $G_p \cdot D_p$ would equal to I_p. However, the computed result D_p does not reflect the real world situation. In reality, the number of nodes we can deploy at a place must be non-negative integers, while the results computed from Equation (9) can be any real numbers, including negative values. Therefore, we can not directly use the result D_p as the deployment strategy. However, the result D_p is a good heuristics to decide where to deploy sensors. In the result D_p, a larger numeric value requires a higher number of nodes to be deployed to satisfy

the requirement. Note that the number of nodes required can be any real number. We perceive the location which corresponds to the maximum value in D_p as the location that contributes the most in satisfying the detection requirement if a sensor is deployed at the location. When we decide the location to deploy the next sensor, the most contributing location is our first choice.

Procedure $DIFF\_DEPLOY(G_p, I_p, N)\{$
         //$N = max(U, V)$
   $invG_p = inv(G_p)$; //Get the inverse matrix of G_p
   $remainI_p = I_p$;
   $D_p = zeros(N \times N, 1)$; /*$D_p$ is a $(N \times N) \times 1$
         matrix, initialized with all zeros.
   $while\ sum(remainI_p) < 0$ /*there are still some
         locations whose miss probability
         thresholds are not satisfied*/
     $nextD_p = invG_p \times remainI_p$;
     Find the maximum value $nextD_p(k)$ in
     $nextD_p$, which satisfies the following
     constraints: $remainI_p(k) < 0\&\&D_p(k) == 0$;
     $D_p(k) = D_p(k) + 1$; /*Place the next sensor
         at location k*/
     $remainI_p = I_p - G_p \times D_p$; //Update $remainI_p$
     Set any positive value in $remainI_p$ to zero;
   end
$\}$

Fig. 2.   DIFF_DEPLOY node deployment algorithm.

DIFF_DEPLOY makes use of the heuristics provided by the result D_p. Figure 2 shows the pseudo code of DIFF_DEPLOY. The algorithm runs iteratively and places one sensor in the sensor field at a time. During each iteration, $nextD_p$ is computed based on the remaining requirement $remainI_p$, which is the difference between I_p specified in the requirement and the logarithmic collective miss probability distribution provided by the nodes that have already been deployed. The most contributing location based on the remaining requirement $remainI_p$ is used as the location to place the next node, which corresponds to the maximum value in $nextD_p$. Then the remaining requirement $remainI_p$ is updated based on the newly deployed node. The algorithm terminates when all the elements in $remainI_p$ are zero, which means that the miss probability thresholds at all the locations are satisfied.

VII. MIN_MISS ALGORITHM AND UNCERTAINTY

In this section, we briefly describe the MIN_MISS deployment algorithm presented in [10], which to the best of our knowledge is the state-of-the-art node deployment algorithm based on probabilistic detection model. Although MIN_MISS achieves better performance than the random node deployment algorithm, we show that in Section VIII our algorithm DIFF_DEPLOY achieves much better performance than MIN_MISS, especially for differentiated detection requirements. For brevity, we use RANDOM to denote the random node deployment algorithm.

In MIN_MISS, for each iteration, the algorithm decides the location for the next node to be deployed. We call each location at which no sensor node is deployed a "candidate location". If we deploy a new node at candidate location (i, j), the collective miss probability at location (x, y) is then $p_{miss}((x, y), (i, j)) \times M(x, y)$. The overall miss probability $M_{overall}(i, j)$ is defined as the sum of the collective miss probabilities at all possible locations when the new node is deployed at location (i, j), or

$$M_{overall}(i, j) = \sum_{(x,y) \in Grid} p_{miss}((x, y), (i, j)) \times M(x, y). \quad (10)$$

In MIN_MISS, candidate location (i, j), which has the minimum $M_{overall}(i, j)$ among all the candidate locations, is selected for the next sensor to deploy. The algorithm terminates when the miss probability of each grid point is smaller than or equal to its miss probability threshold.

The paper [10] states that MIN_MISS is an "uncertainty-aware sensor node placement algorithm". The uncertainty here means that when we deploy sensors, sensors may not be placed at the exact intended positions because of uncontrollable conditions, such as localization error, and drifting of sensors in undersea sensor networks due to water flows. By incorporating the uncertainty in the node deployment, the node detection model is modified. However, it does not affect the deployment algorithm. We show it based on the equations from [10].

In [10], a Gaussian probability distribution is used to model the deviation between the intended sensor locations and the sensors' actual locations. In this Gaussian probability distribution model, the intended location (x, y) is the mean value and the standard deviations are σ_x in the x dimension and σ_y in the y dimension. With this model, the probability of a node intended to be deployed at location (x, y) but actually goes to location (x', y') is:

$$p_{deploy}((x, y), (x', y')) = \frac{e^{-\frac{(x'-x)^2}{2\sigma_x^2} - \frac{(y'-y)^2}{2\sigma_y^2}}}{2\pi\sigma_x\sigma_y} \quad (11)$$

Let $Area$ denote all the possible locations a sensor can be deployed due to uncertainty. Then the possibility of a target at grid point (m, n) being detected by a sensor which is intended to be deployed at location (x, y) can be denoted in Equation (12):

$$p_{uncertainty}((m, n), (x, y)) = \frac{\sum_{(x',y') \in Area} p((m, n), (x', y')) p_{deploy}((x, y), (x', y'))}{\sum_{(x',y') \in Area} p_{deploy}((x, y), (x', y'))} \quad (12)$$

As shown in Equation (12), by introducing uncertainty, the node detection model changes from $p((m, n), (i, j))$ to $p_{uncertain}((m, n), (i, j))$. We call the new model that considers uncertainty $p_{uncertain}((m, n), (i, j))$, the uncertainty aware node detection model. Figure 3 shows the uncertainty

aware node detection model. We assume that σ_x equals σ_y, which is denoted as stdev in the figure. The Gaussian model is truncated and we set the maximum distance error between the intended location and the actual location to 3. We set a to 0.5 to compute the corresponding detection probabilities. As shown in Figure 3, a larger stdev significantly decreases the detection probability near the intended sensor location, but slightly increases the detection probability far away from the intended sensor location.
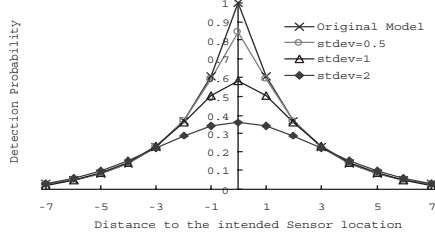
Fig. 3.   Sensor detection model when uncertainty is considered.

Since MIN_MISS is a general node deployment algorithm as DIFF_DEPLOY, which is not designed just for one specific node detection model, it makes more sense to decouple this uncertainty from the node deployment algorithm. When uncertainty is introduced into the node deployment, we only need to feed the uncertainty aware node detection model, instead of the original node detection model, into the node deployment algorithm. We give some experimental results in Section VIII-C when uncertainty in deployment is considered, to show the robustness of our proposed algorithm.

VIII.   Performance Evaluation

We simulate DIFF_DEPLOY, MIN_MISS, and RANDOM in Matlab and compare their performance. In our simulations, we set the grid dimension to 50×50. All the points in RANDOM are the mean values computed from 20 trials. Unless otherwise specified, the sensing range is set to 7; the value of a is set to 0.5.

For this performance evaluation, four sets of experiments are designed. In the first set, we compare the performance of different deployment algorithms for uniform detection requirements. In the second set, the performance of different deployment algorithms for differentiated detection requirements is evaluated. In the third set, we investigate the performance of different deployment algorithms when uncertainty in the node deployment is introduced. In the forth set, we compare the performance of DIFF_DEPLOY to the optimum for small scale problems.

A. Evaluation Set 1: Uniform Detection Requirements

We first study the performance of DIFF_DEPLOY, MIN_MISS and RANDOM when the miss probability thresholds at all locations are the same.

Figure 4 shows the performance of different deployment algorithms when we change the miss probability threshold.
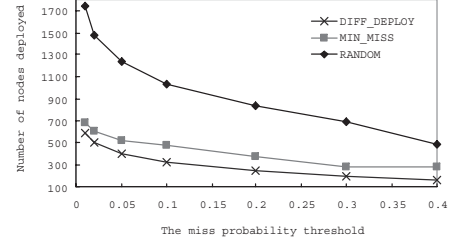

Fig. 4.   Performance of DIFF_DEPLOY, MIN_MISS and RANDOM with various miss probability thresholds for uniform detection requirements.

As shown in Figure 4, the number of nodes decreases as the miss probability threshold increases in all three algorithms. Both MIN_MISS and DIFF_DEPLOY use much fewer nodes than RANDOM for uniform detection requirements. However, DIFF_DEPLOY achieves better performance than MIN_MISS even for uniform detection requirements. In DIFF_DEPLOY when we deploy a new sensor, both the remaining miss probability threshold distribution and the node detection model are directly integrated into the computation as shown in Equation (9). The performance advantage of DIFF_DEPLOY over MIN_MISS is more significant when the miss probability threshold is larger. Overall, DIFF_DEPLOY uses 14%-42% fewer sensors than MIN_MISS.
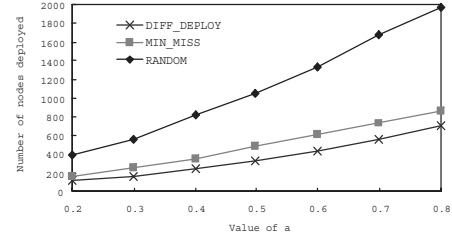

Fig. 5.   Performance of DIFF_DEPLOY, MIN_MISS and RANDOM with various a values for uniform detection requirements.

Figure 5 shows the performance of different deployment algorithms with various a values, when the miss probability threshold is set to 0.1. As shown in Figure 5, the number of nodes deployed increases as the value of a increases, because the larger the value of a is, the faster the detection probability drops when the distance between the target and the node increases. Both DIFF_DEPLOY and MIN_MISS greatly outperforms RANDOM with all a values. Overall, DIFF_DEPLOY achieves 20%-34% better performance than MIN_MISS.

Figure 6 shows the performance of different deployment algorithms when we change the sensing range. The miss probability threshold is set to 0.1. As shown in the figure, the number of nodes deployed first decreases fast in all three algorithms as the sensing range increases. Then the number of nodes starts to stabilize when the sensing range is above 7, because the detection probability becomes small when the distance between the target and the sensor is large. Overall,
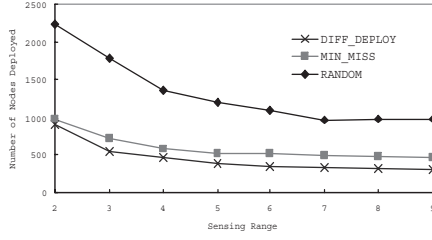
Fig. 6. Performance of DIFF_DEPLOY, MIN_MISS and RANDOM with various sensing ranges for uniform detection requirements.
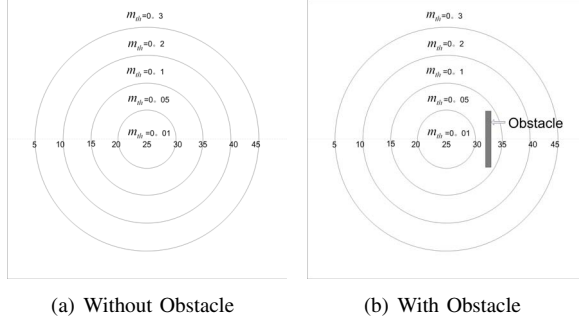

(a) Without Obstacle   (b) With Obstacle

Fig. 7. Miss probability threshold distribution.

DIFF_DEPLOY uses 9%-34% fewer sensors than MIN_MISS with various sensing ranges.

B. Evaluation Set 2: Differentiated Detection Requirements

We study the performance of DIFF_DEPLOY, MIN_MISS and RANDOM when the specified miss probability thresholds at different locations are different in this section. Figure 7(a) shows the specification of the miss probability threshold distribution of the field. Figure 8 and Figure 9 show the results of performance evaluation when there are no obstacles in the environment. Then we show their performance when there are obstacles in the environment in Figure 10.
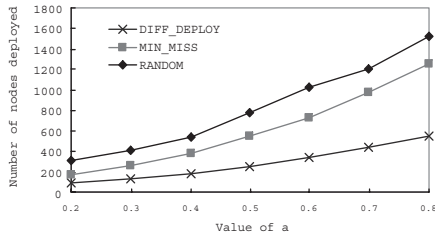

Fig. 8. Performance of DIFF_DEPLOY, MIN_MISS and RANDOM with various a values for differentiated detection requirements.

Figure 8 shows the performance of the three deployment algorithms with various a values. When the miss probability thresholds at different locations are different, the performance advantage of MIN_MISS over RANDOM is smaller than that when the miss probability thresholds at all locations are the same, because MIN_MISS does not take into consideration

the issue of different miss probability thresholds at different locations. The performance advantage of DIFF_DEPLOY over MIN_MISS is more significant for differentiated detection requirements. By adapting to the different miss probability thresholds at different locations, DIFF_DEPLOY uses 47%-57% fewer nodes than MIN_MISS.
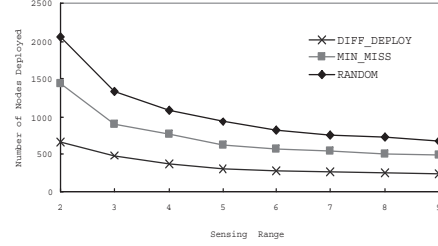

Fig. 9. Performance of DIFF_DEPLOY, MIN_MISS and RANDOM with various sensing ranges for differentiated detection requirements.

Figure 9 shows the performance of DIFF_DEPLOY, MIN_MISS and RANDOM with various sensing ranges. With different sensing ranges, the performance advantage of DIFF_DEPLOY over MIN_MISS is also much more significant for differentiated detection requirements. The number of nodes used in DIFF_DEPLOY is 50%-54% fewer than that used in MIN_MISS.
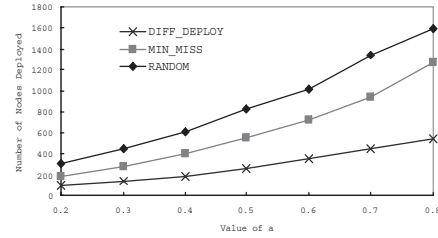

Fig. 10. Performance of DIFF_DEPLOY, MIN_MISS and RANDOM with various a values for differentiated detection requirements.

DIFF_DEPLOY shows promising performance for differentiated detection requirements when there are no obstacles. We investigate the performance of DIFF_DEPLOY when there are obstacles in the environment. The distribution of the obstacle is shown in Figure 7(b). We assume for simplicity that a sensor can not detect a target if there is an obstacle lying between the sensor and the target. Thus, the nodes deployed near the obstacle have different detection models than others. We show the performance of the three deployment algorithms in Figure 10 when there are obstacles. Compared to the performance shown in Figure 8 when there are no obstacles, each deployment algorithm uses a little larger number of nodes. The existence of obstacle restricts the sensing ability of the nodes deployed close to it, which results in more nodes. Overall, DIFF_DEPLOY requires 47%-54% less nodes than MIN_MISS with various a values when there are obstacles.

C. Evaluation Set 3: Uncertainty

In this section, we show the performance of the three deployment algorithms when uncertainty in the node deployment is considered. In this set of experiments, we use the same miss probability threshold distribution as that shown in Figure 7. Note that by considering uncertainty, it only changes the node detection model. It means that in DIFF_DEPLOY, only Y_p needs to be replaced with the one that is constructed based on the uncertainty aware node detection model.
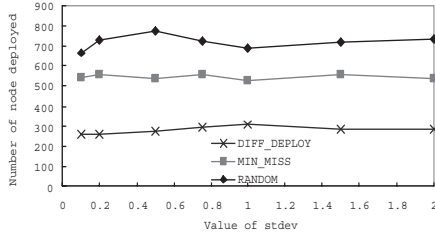

Fig. 11. Performance of DIFF_DEPLOY, MIN_MISS and RANDOM when uncertainty is considered.

Figure 11 shows the performance of the three deployment algorithms when we change the stdev values. In this set of experiments, the maximum distance error between the intended location and the actual location is set to 3. With uncertainty considered, DIFF_DEPLOY still maintains great performance improvement over MIN_MISS. Overall, DIFF_DEPLOY uses 42%-54% fewer sensors than MIN_MISS with various values of stdev, when uncertainty is considered.
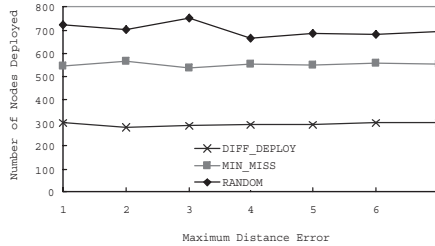

Fig. 12. Performance of DIFF_DEPLOY, MIN_MISS and RANDOM when uncertainty is considered.

Figure 12 shows the performance of the three deployment algorithms when we change the maximum distance error between the intended location and the actual location. The stdev is set to 2 in this set of experiments. As shown in the figure, by varying the maximum distance error, the performance of each algorithm does not change much. Overall, DIFF_DEPLOY uses 45%-51% fewer sensors than MIN_MISS with different maximum distance errors.

D. Evaluation Set 4: Towards Optimum

We have shown the performance superiority of DIFF_DEPLOY over MIN_MISS in various settings. In this section, we compare the performance of DIFF_DEPLOY

to the optimum for small scale problems, because it is computationally infeasible to obtain the optimum when the input size is large. For this set of experiments, the miss probability thresholds at different locations are the same; the grid dimension is set to 5×5; the sensing range is set to 5; the value of a is set to 0.5. We use OPTIMUM to represent the optimal node deployment algorithm, which always uses the minimum number of nodes to satisfy the requirements.
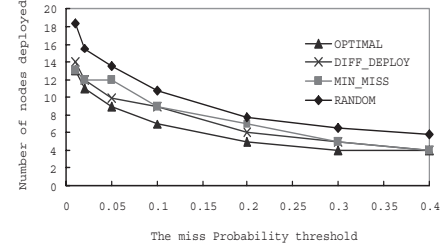

Fig. 13. Performance of OPTIMUM, DIFF_DEPLOY, MIN_MISS and RANDOM.

Figure 13 shows the number of nodes needed for OPTIMUM, DIFF_DEPLOY, MIN_MISS and RANDOM when the miss probability threshold varies. Overall, for small scale problems, DIFF_DEPLOY is the algorithm which achieves the closest performance to the OPTIMUM, except that when the miss probability threshold is 0.01, DIFF_DEPLOY actually uses one more sensor than MIN_MISS. It is not surprising to see that in some special cases, DIFF_DEPLOY uses a few more sensors than MIN_MISS. Both DIFF_DEPLOY and MIN_MISS are heuristic based algorithms and there is no guarantee that one algorithm always performs better than the other one under all settings. For this small scale problem, DIFF_DEPLOY shows quite close performance to OPTIMUM, which on average uses 1 more sensor than OPTIMUM.

IX. CONCLUSIONS AND FUTURE WORK

In this paper, we focus on differentiated deployment problem, in which the required detection probability thresholds at different locations are different. We apply the probabilistic detection model, which is more realistic than the binary detection model. We show that the relationship between the node deployment strategy and the logarithmic collective miss probability distribution is Linear Shift Invariant (LSI). We formulate the differentiated deployment problem as an integer linear programming problem, which is a well known NP-hard problem. The main contribution of this paper is DIFF_DEPLOY, a differentiated node deployment algorithm that achieves much better performance than the state-of-the-art algorithm MIN_MISS, especially for differentiated detection requirements. DIFF_DEPLOY provides the flexibility to enforce the differentiated detection requirements and at the same time minimize the total number of nodes deployed.

In the future, we will investigate other applications of using the LSI relationship between the node deployment strategy and the logarithmic collective miss probability distribution.

We will also study the possible solutions when the sensing range is not uniform, in which case the sensing ranges in different directions are different. This adds extra complexities into the current problem, which requires us to decide not only the deployment locations of the sensors, but also the direction of each sensor.

REFERENCES

[1] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "An Energy-Efficient Surveillance System Using Wireless Sensor Networks," in *ACM Mobisys*, 2004.

[2] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *ACM WSNA*, 2002.

[3] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho, "Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks," *IEEE Transactions on Computers*, vol. 51, pp. 1448–1453, 2002.

[4] V. Isler, S. Kannan, and K. Daniilidis, "Sampling Based Sensor Network Deployment," in *IEEE/RSJ IROS*, 2004.

[5] K. Kar and S. Banerjee, "Node placement for connected coverage in sensor networks," in *IEEE WiOpt*, 2003.

[6] S. Shakkottai, S. Srikant, and N. Shroff, "Unreliable Sensor Grids: Coverage, Connectivity and Diameter," in *IEEE InfoCom*, 2003.

[7] "http://www.xbow.com/."

[8] R. Stoleru, T. He, and J. Stankovic, "Walking GPS: A Practical Solution for Localization in Manually Deployed Wireless Sensor Networks," in *IEEE EmNetS-I*, 2004.

[9] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, "Autonomous Deployment and Repair of a Sensor Network using an Unmanned Aerial Vehicle," in *IEEE ICRA*, 2004.

[10] Y. Zou and K. Chakrabarty, "Uncertainty-Aware and Coverage-Oriented Deployment for Sensor Networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 788–798, July 2004.

[11] S. Dhillon and K. Chakrabarty, "Sensor Placement for Grid Coverage under Imprecise Detections," in *FUSION*, 2002.

[12] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Networks," in *IEEE InfoCom*, 2001.

[13] Y. Zou and K. Chakrabarty, "Sensor Deployment and Target Localization in Distributed Sensor Networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 1, pp. 61–91, February 2004.

[14] R. Kershner, "The Number of Circles Covering a Set," *American Journal of Mathematics*, vol. 61, pp. 665–671, 1939.

[15] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, A. Tirumala, Q. Cao, T. He, J. Stankovic, T. Abdelzaher, and B. Krogh, "Lightweight Detection and Classification for Wireless Sensor Networks in Realistic Environments," in *ACM SenSys*, 2005.

[16] T. Yan, T. He, and J. Stankovic, "Differentiated Surveillance for Sensor Networks," in *ACM SenSys*, 2003.

[17] A. Elfes, "Occupancy Grids: a Stochastic Spatial Representation for Active Robot Perception," in *UAI*, 1990.

[18] H. C. Andrews and B. Hunt, "Digital Image Restoration," *Prentice Hall, Englewood Cliffs, New Jersey*, 1977.

[19] B. Hunt, "A Matrix Theory Proof of the Discrete Convolution Theorem," *IEEE Transactions on Audio Electroacustics*, vol. AU-19, pp. 285–288, 1971.